

BulletsOP 2.0

by: Milan Suk

v.2.0.10



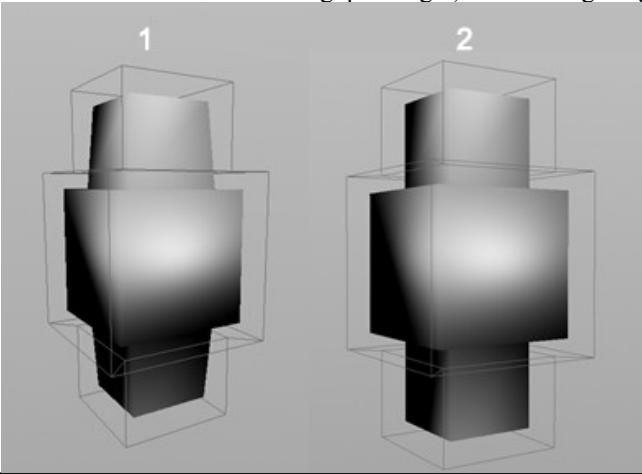
btPrepare

Adds necessary initial data to input geometry.

Inputs

Source	Geometry to which attach initial data
Previous prepare data	Data from previously created btPrepare node

Parameters

Index start	Shifts start value of bt_index attribute
Original geometry	“Off” the output geometry have padding. “On” the output geometry will be the same as original (that came to Source input) + geometry with padding
Padding	Reduces geometry size by spinner value (similar to scale down each piece) to avoid pieces from intersection.
Normal	Different methods for handling padding 1) Normal edge angles 2) Normal planes 
Extra padding	Value that summed with padding. Not changes geometry itself and used only internally by solver.

GROUPS

Group prefix	Group prefix which will be used for assigning bt_index attribute value. For example: if input geometry has groups: Piece01, Piece02, Piece03 and Group prefix set to “Piece” than bt_index has values 1, 2, 3 for corresponding primitives(no attached 2 nd input and “index start” set to 1)
Delete all Groups	Removes all existing groups in output geometry.
Delete only prefix groups	Removes only groups with prefix described in Group prefix

COPIES

Using this method will boost performance if you objects consists of identical copies.

<i>Copy npoints</i>	Total points number of one object copy
<i>Copy nprims</i>	Total primitives number of one object copy

COMPOUND

Proper handling complex objects by representation their forms with more simple shapes

<i>Compound objects</i>	Creates pairs for compound objects. For example if you have 6 objects: A, B , C, D, E, F and value set to 3 it will create "pairs" A-B-C, D-E-F. In simulation you will have 2objects that have 3 collision shapes. (0 means off)
<i>Compound first</i>	“On ” - first object will not be collision shape, but it will be shape for compound(only for rendering) => B-C(A for rendering) E-F(D for rendering) => 2 RBD objects(every has 2 collisions shapes)

DECOMPOSITION

Another method of handling complex objects. Internally represents object by simple convex shapes.

<i>Precise convex</i>	When “off“ uses faster algorithm for checking If object is convex, but not so accurate. “On” – slow but more accurate.
<i>Convex Toler</i>	Precision of convex testing
<i>Tetra</i>	Describes input object shape as summations of tetrahedrons
<i>Precision</i>	Higher values generates more precise shape decomposition.
<i>Voxel</i>	Divides input object shape into voxels.
<i>Cell size</i>	Size of voxels (lower values – better shape representation)
<i>Compound GEO</i>	Adds geometry which is used for generating tetras/voxels to node output - it has bt_index of compound object.



btExtraCells

Groups primitives by specified attributes and synchronizes index of low poly and high poly geometry (see VORONOI example inside MY_BREAK node)

Inputs

Source	Low poly simulation geometry
Unknown source	High poly geometry that gets final transforms

Attribute Name	This attribute will be used to place set of primitives in groups.
Delete Empty	Removes empty cells from input geometry.
Group Name	Group prefix for generated output groups



btBuild

Builds additional data to prepare objects for simulation.

Inputs

Source	Require object after btPrepare
---------------	--------------------------------

Create Group/ Group name	Creates group and add all created points by this node to group.
-------------------------------------	---

RBD's

Hull	Different methods for representing collision shape.
Cog	Different methods for computing objects Center of Gravity.
Static	“On”- object’s behavior will be treated as passive rigid body .
Update every frame	“On” set <i>bt_update</i> attribute to 1 else 0. (Use this option if your’s object have an animated shape and require data updates every frames (in cost of speed).)
Calculate mass from Volume and Density	“On” – objects mass calculated by user’s set density and computed volume.
Density kg/m3	Density value that used to calculate object mass.
Copy primitives attributes	Transfer primitive attributes from input object primitives to output.

CONSTRAINTS

Static

Creates single constraints in space

Constraints	Use “+” “-” to add or remove constraints.
Index Start Object	Index of object FROM which to set constraints.
Index End Object	Index of object TO which to set constraints.
Start/End World Pos	Start/End position of created constraints.
Stiffness	Sets constraints stiffness. With higher stiffness values constraints will be stronger.
Damping	Sets constraints damping. With high values constraints lose more energy over time.
Maximum Force	Set maximum force after which constraint will be broken.
Linear Lock	Sets Lower Limit equal to Upper Limit i.e. locking constraints in space.
Lin Lower Limit	Lower limit of constraint linear movement.
Lin Upper Limit	Upper limit of constraint linear movement.
Ang Lower Limit	Lower limit of constraint angular movement.
Ang Upper Limit	Upper limit of constraint angular movement.
Iteration	Higher values will cause more equations for constraint behavior (in cost of performance). (default “-1” means that solver itself decide how much iterations to solve constraint needed).
Weariness	“off” – if force greater then maximum force constraint will be broken. “on” - sums all forces on constraints from previous frames(saved to bt_max_force attribute).

Generating

Procedurally generates constraints between objects

Maximum distance	Maximum search distance between points for constraints creation.
Stiffness	Sets constraint stiffness. With higher stiffness values constraint will be stronger.
Damping	Sets constraint damping. With high values constraint lose more energy over time.
Maximum Force	Sets maximum force after which constraint will be broken.
Iteration	Higher values will cause more equations for constraint behavior (in cost of performance). (default “-1” means that solver itself decide how much iterations to solve constraint needed)

Weariness	“off” – if force greater then maximum force constraint will be broken. “on” - sums all forces on constraints from previous frames(saved to bt_max_force attribute).
Max Memory(MB)	Maximum memory that constraint creation process can obtain. Higher value can add speed.

Force

Forces tab	Press “+” “-” to add new force.	
Type	Full	Affect both linear and angular velocity.
	Central	Affect only linear velocity.
	Torque	Affects only angular velocity.
	Magnet	Attracts objects to force center.
Position	Force position in space.	
Direction	Force direction. (Keep in mind internally this value will be normalized).	
Scale	Same as strength. Higher values give force more power.	
Object index	Index (bt_index) of objects which force will affects.	
Radius	Limits force influence by radius.	
Fade	Reduces force strength related to distance from center.	

Directory

Attribute Name	Gives possibility to read any data using bt_index attribute. If there are no objects with such index it will return -1.(Use newly created bt_line primitive attribute to access data)
Class	Select which type of attribute you want to read

Emit

Connect Distance	Polygons will marked as “pairs” if the distance between them will be lower than this value.
Break Distance	If distance between polygons will be greater than this value polygons will become emitters



btModify

Sets/ modifies different bullet sop attributes assigned previously .

RBD, Force Group	RBD/Force group in which properties need to be change.
Constraint Group	Constraint group in which properties need to be change.

RBD's

<i>Index</i>	bt_index-> \$BI	Unique index of simulated object
<i>Compound Index</i>	bt_cindex ->\$BCI	Unique compound index
<i>Hull Type</i>	bt_type->\$BTTYPE	Object collision shape type
<i>Translate</i>	P -> \$BT(X,Y,Z)	Position
<i>Rotate</i>	bt_r ->\$BR (X,Y,Z)	Rotation
<i>COG</i>	bt_cog-> \$BCOG(X,Y,Z)	Center of Gravity.
<i>COG rotation</i>	bt_cog_r ->\$BCOGR(X,Y,Z)	Rotation Center of Gravity.
<i>Mass</i>	bt_mass->\$BMASS	Object's mass.
<i>Restitution</i>	bt_restitution->\$BREST	Object's bounciness.
<i>Friction</i>	bt_friction ->\$BFRI	Friction between objects (Friction is the force resisting the relative motion of solid surfaces sliding against each other)
<i>Padding</i>	bt_padding ->\$BPAD	Reduce geometry size (similar to scale down each piece) to avoid pieces from intersections.
<i>Linear Velocity</i>	bt_v->\$BV(X,Y,Z)	Linear velocity.
<i>Angular Velocity</i>	bt_av->\$BAV(X,Y,Z)	Angular velocity.
<i>Linear damping</i>	bt_lin_damp->\$BLDMP	Represents how much linear movement of object will slowdown with time
<i>Angular damping</i>	bt_ang_damp->\$BLDMP	Represents how much rotation movement of object will slowdown with time
<i>Box Size</i>	bt_boxsize->\$BB(X,Y,Z)	Object's bounding box.
<i>Sleeping</i>	bt_sleeping->\$BSLEEP	Sleeping objects ignored by solver until they "wake up" by collision with another objects.
<i>Update</i>	bt_update->\$BUPDATE	Solver will update objects attributes every frame.

Constraints

<i>Index Start Object</i>	bt_index_a-> \$BIA	Index of object FROM what constraint created.
<i>Index End Object</i>	bt_index_b-> \$BIB	Index of object TO which constraint created.
<i>Stiffness</i>	bt_stiffness->\$BSTIFF	Set constraint stiffness. With higher stiffness values constraint will be stronger.
<i>Damping</i>	bt_damping->\$BDAMP	Set constraint damping. With high values constraints lose more energy over time.

Maximum Force	bt_maxforce->\$BMAXF	Maximum force after which constraint will be broken.
Linear Lower Limit	bt_linlowerlimit-> BLL(X,Y,Z)	Sets linear lower limit.
Linear Upper Limit	bt_linupperlimit-> BLU(X,Y,Z)	Sets linear upper limit.
Angular Lower Limit	bt_anglowerlimit-> \$BAL(X,Y,Z)	Sets angular lower limit.
Angular Upper Limit	bt_anglowerlimit-> \$BAU(X,Y,Z)	Sets upper lower limit.
Iterations	bt_iter->\$BITER	Constraint solving iterations.
Weariness	bt_weariness->\$BWEAR	“off” – if force greater then maximum force constraint will be broken. “on” - sums all forces on constraints from previous frames(saved to bt_max_force attribute).
Update	bt_update->\$BUPDATE	Updates constraint properties every frame.
Delete Duplicity	-	Deletes duplicate constraints(“bt_index_a” – “bt_index_b” pair)
Linear Lock	-	Sets Lower Limit equal to Upper Limit i.e. locking constraints in space.

Forces

Type	bt_type-> \$ BTYPE	Force type (Full(0), Central(1), Torque(2), Magnet(3)).
Position	P->\$BT(X,Y,Z)	Force center position.
Direction	bt_dir->\$BDIR(X,Y,Z)	Force direction.
Scale	bt_scale->\$BSCALE	Force strength.
Object index	bt_index->\$BI	Object ID to what force will be applied.
Radius	bt_rad->\$BRAD	Limits force influence by radius.
Fade	bt_fade->\$BFADE	Reduces force strength related to distance from center.

Delete Duplicity	If two objects have many collision points, this option eliminates them to have only one per object.
-------------------------	---

Emits

Connect Distance	bt_emit_connect_dist-> \$BCD	Polygons will marked as “pairs” if the distance between them will be lower than this value.
Break Distance	bt_break_connect_dist->> \$BBD	If distance between polygons will be greater than this value polygons will become emitters
	bt_emit_frame-> \$BF	On which frame geometry will start to emit

Emit frame		
-------------------	--	--



btLoader

Loads data from previous frame from the specified btSolver

Solver Path	Specify from what Bullet Solver you want to load data
Load Geometry	Loads geometry data
Load Properties	Load Properties data
Load Forces	Load Forces data
Load Constraints	Load Constraints data
Load Impacts data	Load Impacts data



btTransform

Moves objects geometry specified in the second input using data from first input

Inputs

Physics Properties	Require data from btSolver to move objects.
Shapes(Geometry)	Sets geometry you want to move.

Output	Prefer Original	If input includes both Original/Collision data it output only Original geometry, otherwise - Collision geometry.
	Original	If input includes both Original/Collision data it shows only Original geometry, otherwise output is empty.
	Collision	Shows only Collision geometry
	Both	Shows Original/Collision geometry
Delete Geometry without properties	Removes geometry that not have any data attached	
Delete SubCompound Objects	Removes decomposition shapes (see btPrepare for details)	
Delete Index Attribute	Removes bt_index attribute from output geometry	
	Adds point velocity attribute.	

Add point velocity	
Update emit frame(not transform)	Updates emit frames for each object (bt_emit_frame primitive attribute) but ignores transform updates(1st input)
Extract Emit Poly	Extracts polygons that marked as emitters
Max Frame Age	Sets how much frames geometry serve “as emitters”



btSolver

The “Core” of physical equation

Inputs

Physics Properties	Require physics properties data
Shapes(Geometry)	Require geometry data
Constraints	Require constraints data
Force	Require force data

Basic

Solver Path	Path to another btSolver
Reset Frame	Frame at which solver will resets all input data
Solve on same frame	Useful if you need to run solver more than once on same frame - for e.g.: inside FOREACH Sop node
Substeps	With higher substeps equation will be more precise in cost of speed
Fps	Frames per second (By default \$FPS expression is used)
Gravity	Gravity force (<0, -9.81, 0> by default)
Ground Plane	“On” – the scene will have invisible static collision ground(y = 0) plane
Collision Group Name	Prefix of collision groups that solver will use for equation.
	“off” - computes force data(radius, ...) on first substep, then applied same data every substep

<i>Force substepping</i>	(not accurate, but faster) “on” - recomputes forces information every substep (more accurate but slow)	
<i>Constraint Substepping</i>	“On” - Solver will delete constraints that have force values greater than Max Force in every substep.	
<i>Output</i>	Full	Output all attributes (bigger cache files)
	Transform	Output transform attributes
	Transform Velocity	Output transform and velocity attributes
	Instance	Output instance attribute (N, Up attributes)
	Zero	Output nothing
<i>Delete Subcompounds Objects</i>	Removes helpers objects that describe objects shape in decomposition mode	
<i>Free world in Extra Thread</i>	If you reset scene, it needs to delete all data which can be slow for large scene. If it turns on, it's deleting old scene in separate thread so you can work on and simulate new one.	